Autonomous flying cameraman with embedded person detection and tracking while applying cinematographic rules

Dries Hulens, Toon Goedemé
Faculty of Engineering
KU Leuven
Sint-Katelijne-Waver, Belgium
dries.hulens@kuleuven.be, toon.goedeme@kuleuven.be

***Abstract*—Unmanned Aerial Vehicles (UAVs) enable numerous applications such as search and rescue operations, structural inspection of buildings, crop growth analysis in agriculture, performing 3D reconstruction and so on. For such applications, currently the UAV is steered manually. However, in this paper we aim to record semi-professional video footage (e.g. concerts, sport events) using fully autonomous UAVs. Evidently, this is challenging since we need to detect and track the actor on-board a UAV in real-time, while automatically – and smoothly – controlling the UAV based on these detections. For this, all four DOF (Degrees of freedom) are controlled in separate simultaneous control loops by our vision-based algorithms. Furthermore cinematographic rules need to be taken into account (e.g. the *rule of thirds*) which position the actor at the visually optimal location in the frame. We extensively validated our algorithms: each control loop and the overall final system is thoroughly evaluated with respect to both accuracy and control speed. We show that our system is able to efficiently control the UAV such that professional recordings are obtained.**

***Keywords*-Autonomous UAV, Cinematographic rules, Person detection and tracking**

I. Introduction

Filming sport events, festivals and even professional movies with UAVs (Unmanned Aerial Vehicle) is becoming increasingly popular the last few years due to the relatively low cost of these UAVs. Furthermore, specific difficult situations like hiking, bicycling or a car pursuit can be captured with a single UAV, while currently cameras are mounted on a special rig, a car or a helicopter for these purposes. However, the disadvantage of these UAVs is that now, besides the cameraman, an additional pilot is needed to control the UAV. This turns out to be a challenging task since the pitch, roll, yaw and altitude should be controlled simultaneously to maintain a perfect shot, especially when no actively steered gimbal is used as in our setup. Additionally, the pilots need to have knowledge of cinematographic rules to ensure that the scenes are visually attractive to the audience. Thus, complying with these specific rules while maintaining control over the UAV makes flying even more challenging. Such rules are for example *the rule of thirds* – i.e. the actor should be positioned at $1/3rd$ such that the action can take place on the remaining $2/3rd$ – and *headroom* – i.e. there should be some room above the head of a person and the top of the frame as seen in Figure 1.

To cope with these challenges we developed an embedded vision-based system which controls the four DOF of the UAV fully autonomously. As such, our system replaces the pilot (the UAV is controlled automatically) and only a director is needed which can maximally focus on giving high-level instructions (e.g. type of shot). For this, we employ computer vision techniques (a person detector and tracker together with distance- and angle-estimation) to autonomously control the UAV. This enables the UAV to automatically follow a person while maintaining a certain shot (e.g. frontal shot, profile shot, close-up) and complying with the cinematographic rules. Since all processing is performed on-board the UAV, a pilot and/or a cameraman are superfluous, thus resulting in an autonomous flying cameraman. However, typical computer vision algorithms rely on high-end hardware to achieve real-time performance (e.g. workstations or computer clusters). Evidently, employing such hardware under a UAV is infeasible. Because the communication latency of an off-board image processing solution would render real-time control loops infeasible, we specifically target on-board processing to ensure instant corrections and full independence of the UAV. Therefore, one of the challenges of this work is to achieve real-time behaviour of these algorithms on light-weight hardware.

The main contributions of this paper are:

- Real-time embedded person detection and tracking to control the relative position of the UAV
- Gaze angle and distance estimation of the person
- Real-life experiments in-the-wild, both technical as well as on aesthetic quality of the produced video

To evaluate our system, as UAV we used the Matrice M100 from DJI. We equipped this UAV with a ZED stereo camera from StereoLabs and a Brix Intel I7 processing board as seen in Figure 2. The stereo camera captures frames of 640×480 pixels at a framerate of 30 FPS. These images are used as input for our processing board, which evaluates each of these images, and generates control signals which are passed to the flight controller of the UAV. Of course,

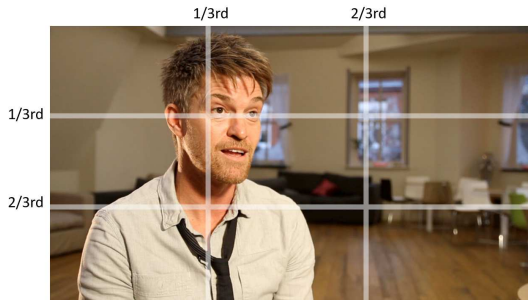

Figure 1. The rule of thirds: When the face is looking to the right, the head should be positioned on 1/3rd on the left of the image and vice versa. Image source: http://www.videoknowhow.co.uk

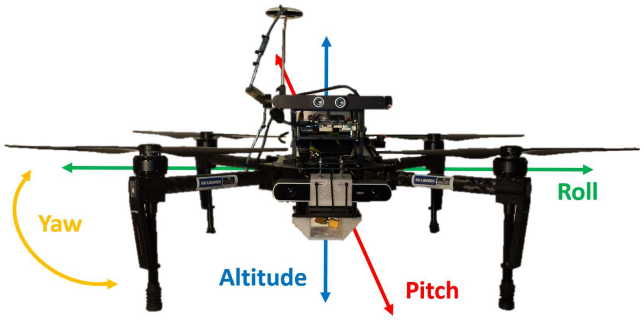

Figure 2. Matrice M100, ZED camera and processing platform, with it's four degrees of freedom that should be controlled

640×480 pixels is not sufficient for professional video footage and a professional camera can be attached to record the actual footage.

The remainder of this paper is structured as follows; In Section 2 we relate our method with the current literature in person detection and tracking with UAVs. In Section 3 we explain how our approach works. In Section 4 our results are discussed and in Section 5 conclusions are drawn and future work is discussed.

II. Related Work

When a UAV is used to autonomously record e.g. a walking actor, three main tasks should be fulfilled:

- The actor should be detected and tracked, such that he can be positioned at $1/3rd$, obeying the *rule of thirds*.
- The UAV should fly at a fixed distance w.r.t. the actor, such that the size of the actor can be made appropriate to the shot type (long shot, mid shot, close-up).
- The UAV should fly under a fixed angle (e.g. profile shot of the face) w.r.t. the actors' face gaze angle.

Most of the current UAV person tracking algorithms are color- or feature-based. In [Lin et al., 2012] a predefined color of the object is used to distinguish the object from the background, followed by template matching to perform detection. In [Haag et al., 2015], [Pestana et al., 2014] a feature-based tracker is used to follow the person with a UAV. Here the person should first be manually selected (using a bounding box) in the life video feed before the tracker can start. Such behaviour could be useful when specific objects of interest (e.g. vehicles) need to be tracked. However, in our case this is a disadvantage since manually selecting the person of interest in every scene is infeasible. We want to follow a human and update/initialize our tracker automatically if the shape or color of the person changes.

Person detection can also be performed using an infrared camera as in [Doherty and Rudol, 2007] where they equipped a large electric helicopter with a heavy infrared camera and high-end processing power. This is infeasible in our case where payload is restricted. Another approach is the use of a model-based person detector. In [De Smedt et al., 2015] an ACF person detector is used to steer the Yaw-axis of the UAV and keep the person centered in the frame. In [Danelljan et al., 2014] they employ a HOG person detector to initialize and update their color-based tracker while the height of the detection bounding box is used as distance measurement to maintain a fixed distance. In [Monajjemi et al., 2016] they use off-board face detection to approach a person and interact with them. Here the height of the face is used as a distance measurement. However, this is not accurate as proven in [Danelljan et al., 2014]. A different approach towards person tracking is found in marker-based systems. Here, a person or actor wears such a visible marker (i.e. QR code) [Vasconcelos and Vasconcelos, 2016] or a GPS bracelet. Evidently, this is certainly not feasible in professional film industry.

While these previously described techniques achieve good accuracy, they cannot be used to record an actor in professional film industry. This is due to poor distance estimation (as we also evaluated in section III-C) and the lack of a cinematographically-aware *shot detector* to position the UAV under a certain angle w.r.t. the actor. In our work we eliminate all these disadvantages and develop a system that meets all three important bullet points given above in an efficient manner. Our system briefly works as follows; First the actor is automatically detected with a DPM [Felzenszwalb et al., 2008] generic person detector that initializes and updates our particle-based color tracker. Next the location of the actor is used to optimally position him in the frame. Simultaneously the distance of the actor w.r.t. the UAV is estimated to maintain a fixed distance by means of a disparity measurement in a stereo camera set-up. Finally, the face-angle of the actor is estimated [Hulens et al., 2016] to position the UAV under the correct angle for a certain shot.

III. Approach

To achieve autonomous flight behaviour of the UAV four DOF need to be controlled as seen in figure 2. When filming

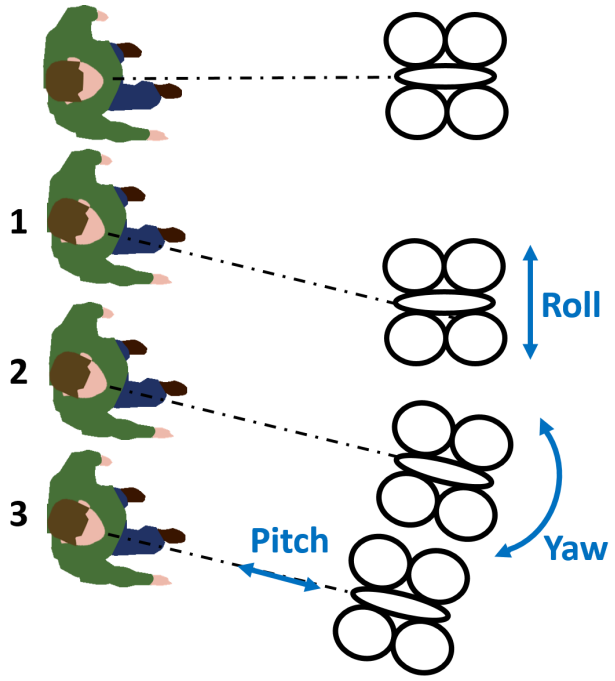

Figure 3. Three degrees of freedom are steered simultaneously to maintain the shot.

an actor these DOF (Pitch, Roll, Yaw and Altitude) should be steered simultaneously to ensure smooth video footage. An example is given in figure 3 where the goal is to take a frontal shot of the actors' head. The top part of this figure displays the initial situation. In the lower parts the actor turns his head, and thus the UAV needs to correct it's position in three simultaneous steps to maintain the frontal shot. For this, the UAV flies to the left (roll) to maintain the gaze direction. As a result of this movement the face is no longer centered and the UAV needs to rotate (yaw) in a clockwise manner to again center the face in the frame. Another consequence of the roll movement is that the distance between the UAV and person increases and the UAV needs to move forward (pitch) to maintain a fixed distance and thus a fixed size of the actor in the video. All three movements are controlled simultaneously. When 1 DOF (e.g. yaw) is controlled to obtain a specific shot, the other DOF (roll and pitch) should also be adjusted to obtain that specific shot. This can not be done automatically, with only information of that 1 DOF (yaw), because the amount of adjustment for the other DOF (roll and pitch) is dependant on the type of shot. Therefore the control loops for each DOF are decoupled as seen in figure 4. This figure displays the overall system which is implemented in ROS (Robot Operating System) and runs in real-time on embedded hardware. Each rounded square represents a ROS node. An additional advantage of decoupling these control loops is that they can be tuned individually. The UAV must for instance not react too quickly upon a sudden rotation of the actors' head but has to react quickly when the distance between UAV and actor becomes smaller. Next each control loop (indicated with different colors) receives the data needed to control the UAV. Furthermore, information is received from the (external) director which still determines the desired position, angle, distance and height of the actor in the frame. In the next subsections each control loop is discussed in detail.

A. *Yaw control*

The yaw rotates the UAV around it's vertical axis and is used to position the actor on the horizontal axis in the frame. The correct position of the actor within the frame depends on the cinematographic *rule of thirds*. To control the yaw and position the actor, we designed a control loop as seen in figure 4 in yellow. The control loop is using four nodes: the Detection node, the Tracking node, the Kalman Filter node and the PID node. To detect the actor (and initialize/update the tracker) we use the C++ implementation of the FFLD person detector of [Dubout and Fleuret, 2012] (fast variant of DPM person detector) and implemented this in ROS. The detector of Dubout uses Fourier transformations to speed up the convolutions between the rescaling operations and filters which are typical for multi-scale person detectors. The person detection node runs at a framerate of 10fps (frames per second). Although the person detector obtains excellent results, it cannot be used without a tracker in this case. First of all our PID control loops should be updated at a minimum of 20Hz for a smooth movement of the UAV and secondly, due to changing variations in appearance a person can have, it's impossible to obtain a 100% reliable detection of the person at each frame. To cope with these variations in appearance we ported a color-based particle tracker from Kevin Schluff[1]. This tracker deals with false or unreliable detections of the actor and predicts the position of the person at 25fps. Each time a person is detected with a high confidence score, the tracker is updated with the position and color histogram of the detection window. Since the output of the tracking node does not have a smooth transition between different cycles, we use a Kalman Filter [Kalman, 1960] to filter the result. When the position of the actor within the frame is determined, the error between the actual position and the desired position (depending on *the rule of thirds*) is calculated and passed to a PID control loop. The latter calculates a smooth control value to steer the yaw axis of the UAV depending on the size of the error and the speed the error changes in time. The yaw controller can now rotate the UAV clockwise or counter-clockwise to position the actor on the horizontal axis in the frame.

[1] https://bitbucket.org/kschluff/particle_tracker

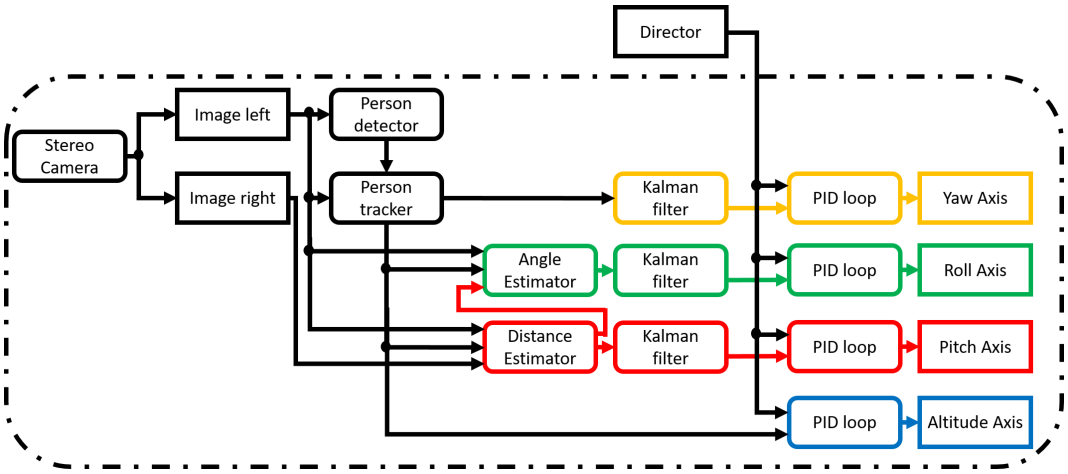

Figure 4. The overall system with four different control loops in color. Each rounded square is a ROS node.

B. Roll control

When the roll is controlled with a certain speed the UAV will start to fly in a circle around the actor since the yaw keeps the actor framed and the pitch ensures a fixed distance between the UAV and actor. As such, changing the type of shot can be achieved by controlling the roll to apply the *the rule of thirds* with different gaze orientations. Evidently, when a shot of a left-looking face is requested by the movie director, the UAV has to know when to stop circling around the person (detect a left-looking face). Hence, we estimate the angle of the face every frame and try to maintain this angle. The latter is done by the Angle Estimator node as seen in figure 4 in green and the result in figure 5 where the angle estimator calculates the angle of a face in three different positions.

This node is based on our previously developed technique [Hulens et al., 2016] to estimate the angle of the face. A face is evaluated using three Viola and Jones [Viola and Jones, 2001] models; a left looking model ($90°$), a right looking model ($-90°$) and frontal looking model ($0°$). The angle of a new input face image is derived as a simple weighted sum of the detection scores. This methods yields an excellent absolute mean error of $13°$ and outperforms others ([Benfold and Reid, 2008], [Rehder et al., 2014]) in accuracy, processing speed and simplicity.

Because of the poor accuracy of the Viola and Jones face detector, the position of the detections is not used to control the yaw. By using a person detector with high accuracy at first, a smaller search region around the detected person is determined for the Viola and Jones face detector which yields less false detections and a higher accuracy of the angle estimator.

As seen in figure 4 (green) the Angle Estimator node receives the coordinates of the actor via the tracker node.

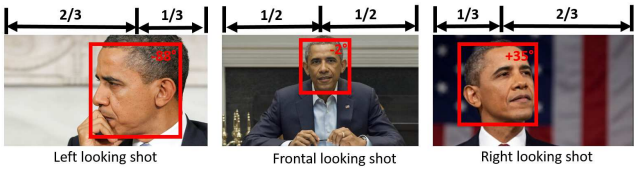

Figure 5. Left: left looking shot, angle is estimated at -88°. Middle: Frontal looking shot, angle is estimated at -2°. Right: Right looking shot, angle is estimated at +35°.

These coordinates are used to determine a smaller search region for the head of the actor in the image. The size of the search region is determined by the size of the actor, which is inversely proportional to the distance (discussed in section III-C). Since the face detection methodology is based on a sliding window approach, searching in a smaller region of the frame is beneficial for processing speed and accuracy. When the entire frame should be used to determine the angle we achieve a maximum framerate of 13FPS while working with a search region yields a framerate of more than 25FPS. When the angle of the face is determined, this value is passed to a Kalman filter to smooth out the result. The kalman filter also ensures that quick rotations of the face doesn't affect the roll movement and the shot. As in the yaw control loop the error is calculated between the current angle of the face and the desired angle (shot determined by director) and passed to a PID controller that controls the roll movement. A consequence of the roll movement is that the distance between the actor and UAV enlarges, this is corrected with the pitch controller in next subsection.

C. Pitch control

A fixed distance between the actor and UAV is maintained by controlling the pitch. To do this automatically we measure the distance between the UAV and actor using a synchronized stereo camera. An alternative would be a Time of flight

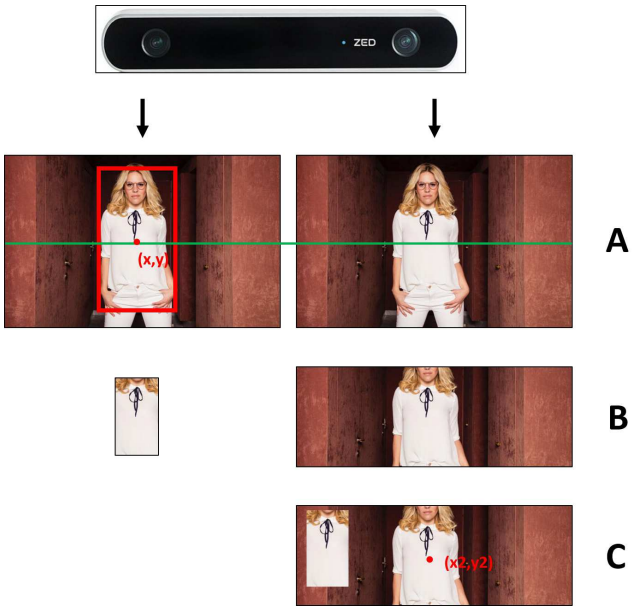

Figure 6. A stereo image is captured by the ZED camera. In the left image a person is detected and a smaller part of the bounding box is used as a template to search the same person in the right image (region around the epipolar line).

or structured light 3D camera, but these active light cameras prove not to be resilient against outdoor ambient light. When a stereo camera is used, a depth map could be constructed that outputs an image where every pixel represents the corresponding distance on that location. However, constructing a depth map is computationally intensive and therefore we only calculate the distance at one location, the position of the actor (estimated by the person detector). For this we need the disparity of the actor, i.e. it's position differs in the left and right image. The person detector can be ran on both the left and right frame yielding two coordinates that can be further used to calculate the distance. However, the latter approach uses a lot of processing power (the overall speed of the person detector will be twice as slow and the result would not be that accurate because we know the detected position can easily be off by a few pixels). Accordingly we detect the actor in the left image and use *template matching* to find the corresponding location of the actor in the right image, which is less processing-power consuming and more accurate. Our Distance Estimator node runs at an average of 25 FPS depending on the distance between actor and UAV as will be explained later.

As in figure 6 (A), we first receive the left and right image from the synchronized camera pair, together with the detection/tracking bounding box. Secondly in (B) we extract the template out of the left image, which is a smaller region of the detection bounding box. A smaller region of the chest is used such that no background is included. Furthermore a search region is extracted out of the right image that is slightly higher (20 pixels) than the template and as wide as the image (640 pixels). This search region is located on the epipolar line (green in (A)) of the location in the left image (a pixel in the left image will be on the same y-location in the right image). The smaller search region is used to speed up the algorithm. Finally Normalized Cross Correlation is used to locate the template in the search region. Evidently, the bigger the distance between UAV and actor, the smaller the template and search region will be and the faster the template matching will be executed.

When the location of the actor is known in both images the distance can be calculated because the stereo pair is calibrated. The output of the distance estimator is also filtered by a Kalman filter, as in figure 4 (red), to smooth the transitions between measurements and to cope with missing distances (e.g. when the confidence of the template matching is too low). In the latter case the prediction of the Kalman filter is used as a measurement. The error between the measured distance and the required distance is passed to a PID loop that controls the actual Pitch of the UAV.

D. Altitude control

The last degree of freedom is the altitude, using the coordinates of the tracker to position the person by default $1/3rd$ under the top of the frame (*cinematographic head room* at 320 pixels on the vertical axis) or at a different location determined by the director. The actual height and desired height are compared and the error between the two is calculated. This error is passed to a PID loop and a velocity to move up or down is calculated and sent to the flight controller to maintain a fixed altitude as in figure 4 (blue).

E. Embedded implementation

Due to the limited processing power on-board most UAVs, vision algorithms are often ran off-line on a ground station ([Monajjemi et al., 2016], [Pestana et al., 2014]) connected via wifi or Radio Frequency and receiving images from the UAV. Those images are then processed and control commands are send back to the UAV to correct it's position. This way of communication introduces a lot of disadvantages such as limited communication distance and a delay between sending and receiving commands. To conquer these disadvantages we mounted an embedded vision processing platform on the UAV. All images are processed on-board the UAV which makes it completely autonomous. The processing platform we use is a Brix mini computer measuring 10×10 cm with an Intel i7 4770R processor, 4GB RAM and a 120GB Solid State disk. This platform weighs 172 gram and has a power consumption of 26 Watt. All our C++ code is implemented in ROS so that different programs (nodes) can run simultaneously and communicate with each other efficiently. DJI provides ROS nodes to communicate with their flight controller making it easy to set up the communication. Another advantage of ROS is

Figure 7. UAV is controlled to keep the actor in the center of the frame.

that every node can independently be debugged and tested. We use PID controllers to steer the UAVs' flight controller due to their simplicity in trimming compared to e.g. LQR controllers. Furthermore we developed an Android mobile app to tune the PID settings during flight. The smartphone app is connected with the remote controller which sends the data to the flight controller. This data is then passed from the flight controller to the processing platform and used to tune the PID values from the four control loops. The remote controller is only used to connect the Android app with the UAV and to intervene in cases of emergency.

IV. Experiments and Results

To validate our system we first conducted separate experiments on the different controller parts followed by a larger experiment to evaluate the entire system, as well as a subjective system evaluation in the ultimate end result: the aesthetic quality of the produced video.

A. Yaw controller

The yaw controller ensures that the actor is framed on the correct position in the frame, depending on the *rule of thirds* and the wishes of the director. In this experiment we stipulated that the actor should be positioned in the center of the frame at all time (which is as good as any arbitrary position). We recorded 600 frames from a person walking around (played by several different actors), and measured the error between the center of the frame and the actual location of the actor in the frame while the yaw was automatically controlled to keep the actor centered as in figure 7.

Figure 8 shows the cumulative distribution of the error value (i.e. the deviation from the center position in pixels).

An average error of 70 pixels in both directions (i.e. 140 pixels in total) is observed, for an image width of 640 pixels. Keep in mind that this exact error is of less concern in the professional film industry. There, it is more important that the actor is approximately located in the center of the frame. For this, often deviations upto $1/3rd$ of the frame width are allowed (in our case 107 pixels). As such, small variations are no problem. If we want to comply to this rule, in our frames deviations upto 107 pixels in both directions are tolerated. In this case, in 78% of all frames the actor is correctly positioned.

The accuracy might further be increased by adjusting the specific PID parameters. If we allow for a more aggressive adjust of the yaw axis, the actor will be faster positioned

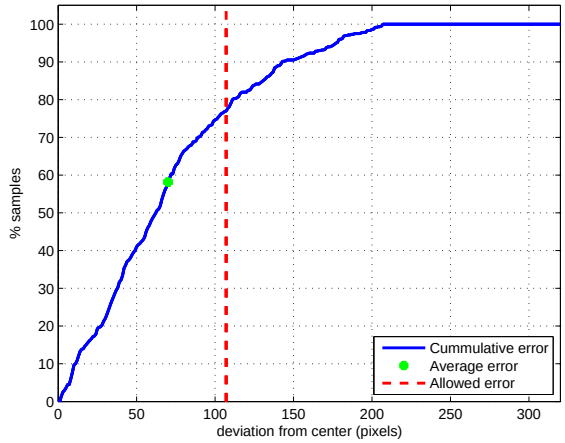

Figure 8. For each error value the percentage of all frames with equal or lower error rate is displayed. The average error is displayed with a green dot. The allowed error is marked with a red line.

Figure 9. 1: Indoor distance test, 2: outdoor distance test.

correctly, although this results in an unnatural jerky camera movement and thus is unpleasant for the audience. We can conclude that the yaw axis is optimally controlled, and this ensures that the actor is always located at the required position in the frame.

B. Pitch controller

The pitch controller ensures a fixed size of the actor in the image by determining the distance w.r.t. the actor. Because most of the pursuit shots are filmed at a distance from $2.5m$ to $4m$, in our experiments we focused on these distances to correctly measure the distance. We conducted both experiments indoors with a non-flying UAV as well as outdoors with a flying UAV. In the indoor experiment we mounted the UAV on a pole and placed markings on the floor at $2.5m$, $3m$, $3.5m$ and $4m$ whereafter we recorded 470 frames of multiple people at the different markings as seen in figure 9 (1).

As a first naive baseline method we estimated the distance using the height of the detection bounding box which is inversely proportional to this distance between the UAV and actor. As seen in figure 10 (red), at each marking we calculated the average distance together with it's standard deviation. When only using the height of the bounding box as distance measurement a mean standard deviation of $35cm$

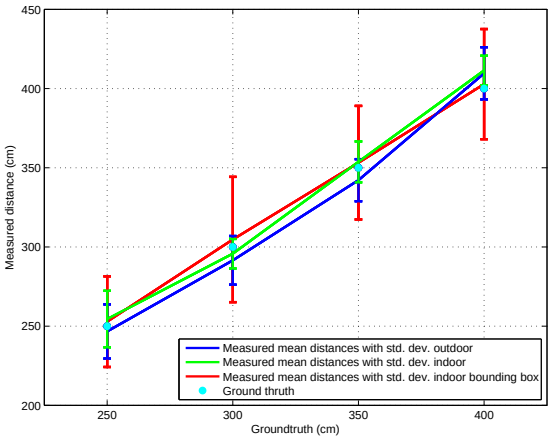

Figure 10. The average measured distance is plotted for each ground truth markings ($2.5m$, $3m$, $3.5m$ and $4m$) together with the standard deviation. Red: baseline. Green: distance estimated with stereo camera-based approach. Blue: same experiment at outdoor location. Cyan: Ground truth markings.

in both directions is observed. Such high deviation is not allowed when flying as close as 2.5m. Therefore, we stepped away from this naive approach, and use a stereo camera-based distance estimation (discussed in section III-C) as seen in figure 10 (green). An average error of $6cm$ is observed with an average standard deviation of $13cm$ in both directions. Additionally, we performed the same experiment outdoor with a flying UAV that was remotely controlled to maintain a fixed position, as seen in figure 9 (2). Again markings were placed at $2.5m$, $3m$, $3.5m$ and $4m$ and distance measures were performed with multiple persons (in 468 frames). In figure 10 (blue), an average error of $8cm$ is observed with an average standard deviation of $16cm$ in both directions. This slightly larger error and standard deviation was predictable due to the fact that a small drift of the UAV from it's fixed position is inevitable. In both indoor and outdoor conditions our stereo camera-based distance estimation achieves an excellent accuracy. Furthermore we proved that our algorithm is able to efficiently measure the distance towards the actor.

C. Roll controller

The shot type is changed by controlling the roll, e.g. from frontal shot to profile shot. In order to evaluate the roll controller we stipulated that a frontal shot should be maintained at all times in this experiment. A test person is asked to turn around it's axis by $\pm 45°$ as in figure 11. Every time the person turns, the measured angle and the time needed for the UAV to correct it's position to maintain the shot is observed. As seen in figure 12, the red lines are the moments the person turns. The blue line is the face angle measured by the UAV and in red the desired angle. As seen,

Figure 11. Sequence 1: Person turns $\pm 45°$ (image 1 and 2), UAV starts to correct it's position (image 3 - 5).

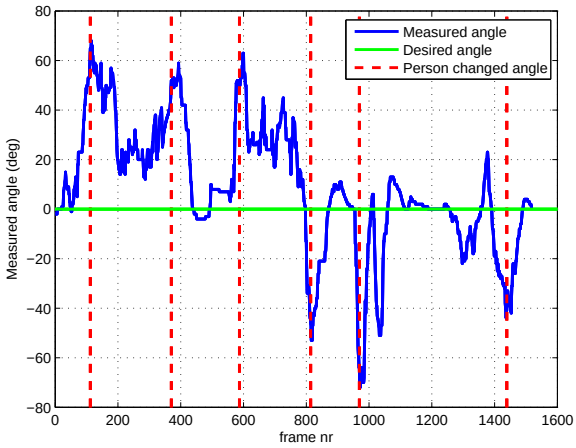

Figure 12. At each red line the test person turns $\pm 45°$ and the UAV starts to correct it's position (angle in blue) to maintain the frontal shot ($0°$ green).

the measured angle changes directly when the person turns. At this point the UAV starts to correct itself by controlling the roll until the angle is back at $0°$. Notice that after the first turn the person did not wait until the UAV was at $0°$ to turn again. And after the $3rd$ turn, it took a while for the UAV to reach $0°$ due to bad lighting conditions (in that sequence) which made it hard to detect the correct angle. The average time needed for the UAV to correct it's position is ± 2 seconds when neglecting the sequence with bad lighting conditions. When the PID is tuned more aggressively, the UAV corrects it's position faster but the movement will be more jerky. We can conclude that the face angle measurement-based roll controller works very well and that a certain type of shot (angle) can be maintained.

D. Overall system

Because in professional video industry accuracy is not as important as viewing experience, the overall system is evaluated on the latter in several experiments where two of them are discussed here. In the first experiment the UAV had to follow a person without taking the angle of the face into account (as in a pursuit from behind). For this we refer to https://youtu.be/kDfnRnxSLuU where the person is detected and tracked (blue and green dot). When the person walks to the left or right, the yaw is readjusted to keep the person centered. If the person walks closer to or farther away from the UAV, the UAV moves backwards or forwards

to keep a fixed distance of $\pm 3.5m$. The second experiment is much more challenging. Here the actor should be captured in a profile shot while walking. Hence, the *rule of thirds* should be taken in consideration. In the second experiment (https://youtu.be/4ZjEJxU3zIA) the person walks to the right and is positioned on $1/3rd$ in the frame while a fixed profile shot is maintained during the recordings. Furthermore, the same sequence was recorded while a professional pilot was flying manually with the UAV whereafter a panel was asked to select the best recordings with viewing experience in mind. Exact 54.5% of the test panel (N = 11) selected the autonomous recordings as the aesthetically better looking. This result indicates that the autonomous recordings are difficult to differentiate from the manual ones, which is the goal of our system. These experiments show that our system can be used to record a pursuit of a walking person fully autonomously without the need of a human cameraman or pilot.

V. Conclusion

In this paper we developed a vision-based control system to steer a UAV so it can film a moving actor from a desired angle fully autonomous. An actor is detected on-board the UAV and it's position in the image, distance and face angle is used to control the four DOF. Furthermore, several cinematographic rules are taken into account to ensure high quality shots. We successfully evaluated each of the four control loops separately for accuracy and speed as well as the overall system for viewing experience. Our system works in real-time and no connection to a ground station is needed. In the future we will experiment with faster tracking and detection algorithms so runners or bicyclists can be followed. Additionally, more cinematographic rules will be implemented so that multiple persons can be followed simultaneously while keeping an aesthetic correct shot.

Acknowledgment

This work is supported by KU Leuven via the CAMETRON project.

References

[Danelljan et al., 2014] Danelljan, M., Khan, F. S., Felsberg, M., Granström, K., Heintz, F., Rudol, P., Wzorek, M., Kvarnström, J., and Doherty, P. (2014). A low-level active vision framework for collaborative unmanned aircraft systems. In *Workshop at the European Conference on Computer Vision*, pages 223–237. Springer.

[De Smedt et al., 2015] De Smedt, F., Hulens, D., and Goedemé, T. (2015). On-board real-time tracking of pedestrians on a uav. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–8.

[Doherty and Rudol, 2007] Doherty, P. and Rudol, P. (2007). A uav search and rescue scenario with human body detection and geolocalization. In *Australasian Joint Conference on Artificial Intelligence*, pages 1–13. Springer.

[Dubout and Fleuret, 2012] Dubout, C. and Fleuret, F. (2012). Exact acceleration of linear object detectors. In *European Conference on Computer Vision*, pages 301–311. Springer.

[Haag et al., 2015] Haag, K., Dotenco, S., and Gallwitz, F. (2015). Correlation filter based visual trackers for person pursuit using a low-cost quadrotor. In *Innovations for Community Services (I4CS), 2015 15th International Conference on*, pages 1–8. IEEE.

[Hulens et al., 2016] Hulens, D., Van Beeck, K., and Goedemé, T. (2016). Fast and accurate face orientation measurement in low-resolution images on embedded hardware. In *Proceedings of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2016)*, volume 4, pages 538–544. Scitepress.

[Kalman, 1960] Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45.

[Lin et al., 2012] Lin, F., Dong, X., Chen, B. M., Lum, K.-Y., and Lee, T. H. (2012). A robust real-time embedded vision system on an unmanned rotorcraft for ground target following. *IEEE Transactions on Industrial Electronics*, 59(2):1038–1049.

[Monajjemi et al., 2016] Monajjemi, V. M., MohaimenianPour, S., and Vaughan, R. T. (2016). Uav, come to me: End-to-end, multi-scale situated hri with an uninstrumented human and a distant uav. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS'16)*.

[Pestana et al., 2014] Pestana, J., Sanchez-Lopez, J. L., Saripalli, S., and Campoy, P. (2014). Computer vision based general object following for gps-denied multirotor unmanned vehicles. In *2014 American Control Conference*, pages 1886–1891. IEEE.

[Vasconcelos and Vasconcelos, 2016] Vasconcelos, F. and Vasconcelos, N. (2016). Person-following uavs. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9. IEEE.

[Viola and Jones, 2001] Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511. IEEE.

[Felzenszwalb et al., 2008] Felzenszwalb, P., McAllester, D., and Ramanan, D. (2008). A discriminatively trained, multiscale, deformable part model. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE.

[Benfold and Reid, 2008] Benfold, B. and Reid, I. (2008). Colour invariant head pose classification in low resolution video. In *BMVC*, pages 1–10.

[Rehder et al., 2014] Rehder, E., Kloeden, H., and Stiller, C. (2014). Head detection and orientation estimation for pedestrian safety. In *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, pages 2292–2297. IEEE.